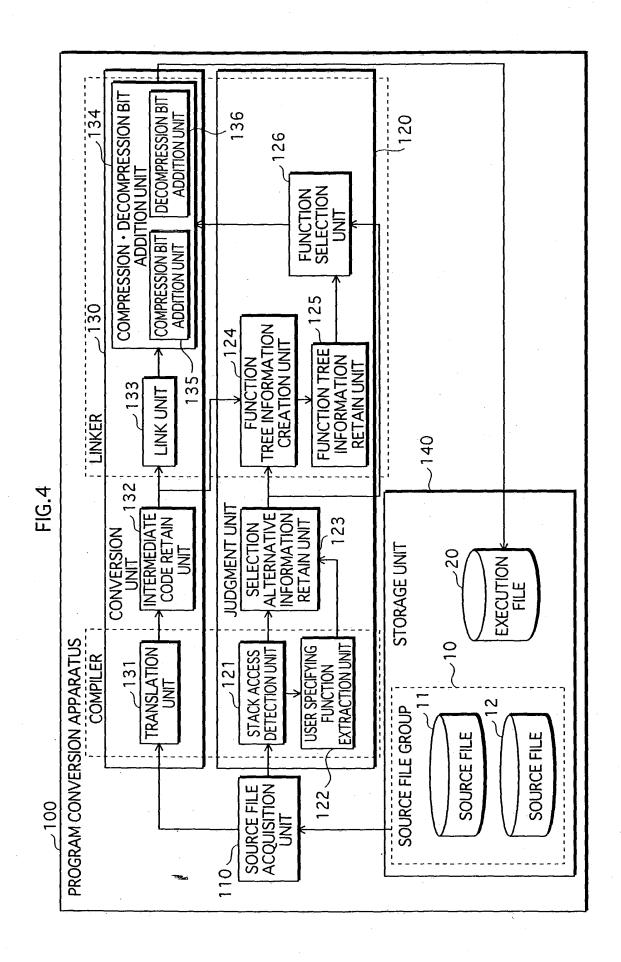
FIG. 1

```
main()
11a <
          func_a();
          func_b();
          func_e();
         func_a()
11b <
           func_c();
         func_b()
11c ~
             if(X){
                    func_c();
             }else{
                    func_d();
          func_d()
 11d 🔾
            asm("load r0,(4,sp)");
asm("and r0,0xf0");
            asm("store (4,sp),r0");
           func_e()
 11e <
           #STACK_COMPRESS
  11f \
           func_f()
  11g <
```

FIG.3

```
12a ____func_c()
{
    func_f();
}
```



123a 123l	1230
FUNCTION NAME	FUNCTION EVALUATION VALUE
func_a	1
func_b	[1
func_c	1
func_d	0
func_e	2
func_f	1.

FIG.6

	HIGHER CORDER	>LOWER ORDER	TREE NUMBER
main-	func_a	— func_c —— func_f	1
	func_b —	func_c func_f	2
<u> </u> 		func_d	3
	func_e		4

FIG.7

125a 125	b 125c
TREE NUMBER	TREE EVALUATION VALUE
1	1 (=1×1×1)
2	1 (=1×1×1)
3	0 (=1×0)
4	2 (=2)

```
Omain
         call_Ofunc_a,[r0:r7],stack compress
         call_Ofunc_b,[r0:r7]
         call_Ofunc_e,[r0:r7],stack compress
302
           ret
        Ofunc_a
           call_Ofunc_c[r0:r7]
           ret,[r0:r7],stack compress
303
        _0func_b
            call_0func_c,[r0:r7]
           else
            call_0func_d,[r0:r7]
           ret,[r0:r7]
         Ofunc_c
           call_0func_f,[r0:r7]
           ret,[r0:r7]
         _Ofunc__d
            load r0,(4,sp)
            and r0,0xf0
            store(4,sp),r0
            ret,[r0:r7]
             ret,[r0:r7]
          Ofunc_e
            ret,[r0:r7],stack compress
 304
          Ofunc_f
             ret,[r0:r7]
```

FIG.9

```
__Omain

__call_Ofunc_a

__ret

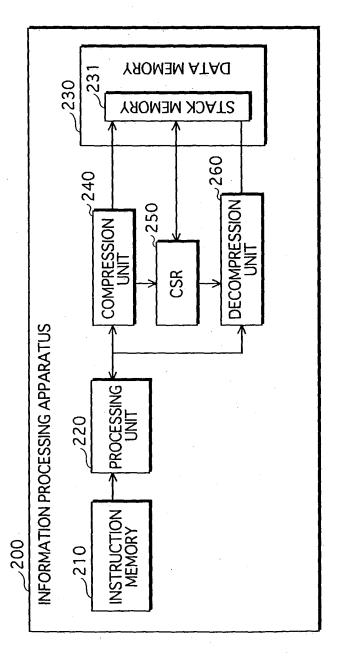
__Ofunc_a

__sotre (sp),[r0:r7],stack compress

__call_Ofunc_c

__load [r0:r7],(sp),stack compress

__ret
```



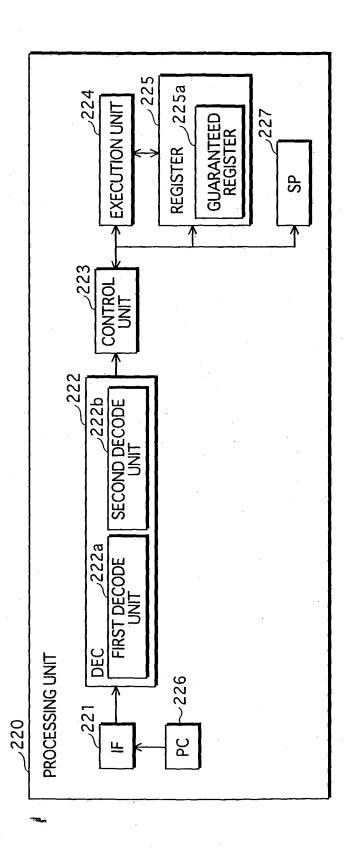


FIG.12

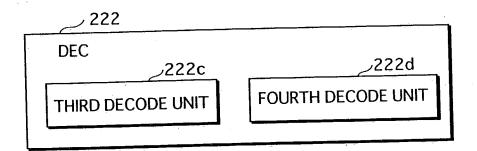


FIG.13

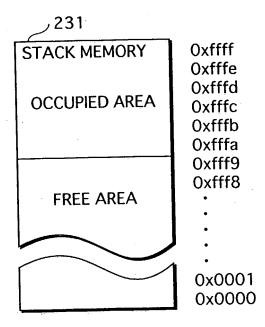


FIG.14

PC

r0:0x80000110

r1:0x00000000

r2:0x0000FFFF

r3:0x00000000

r4:0x80000010

r5:0x80000014

r6:0x50000010

r7:0x50000000

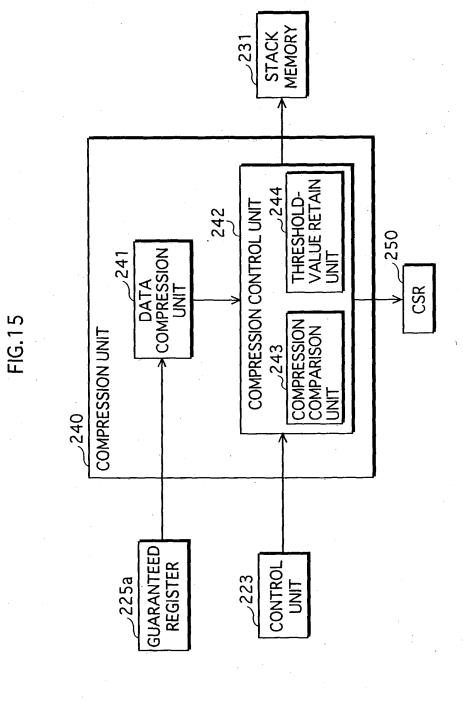


FIG.16

VALUE	HUFFMAN-CODE
0x4	0b00000
0x5	0b00001
0x8	0b0001
Oxf	0b001
Ox1	0b01
0x0	0b1

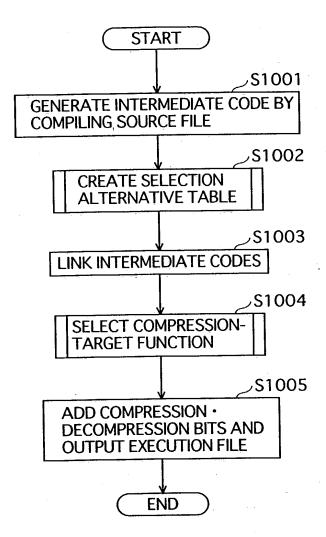
FIG.17

PC 0x1F5FFF92 0x4FF8F08F 0x0007EC3F 0111 CSR

FIG.18

[31]···[8][7][6][5][4][3][2][1][0] 250 0···0 1 1 0 0 0 1 0 1

FIG.20



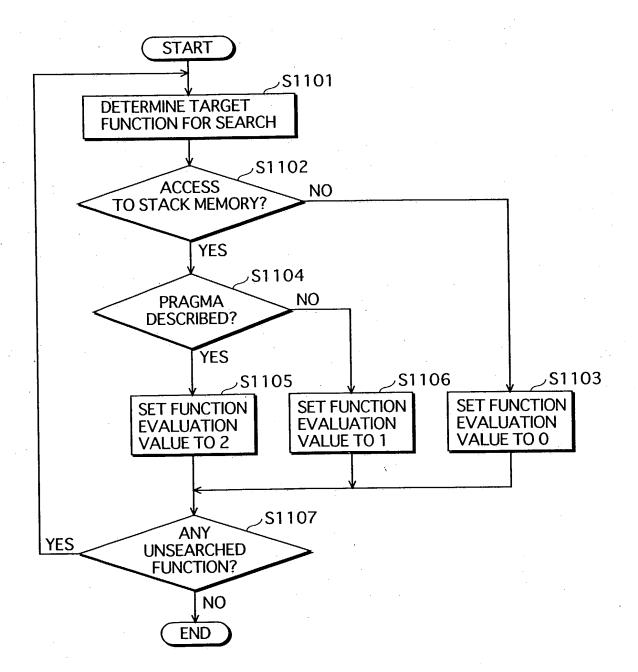
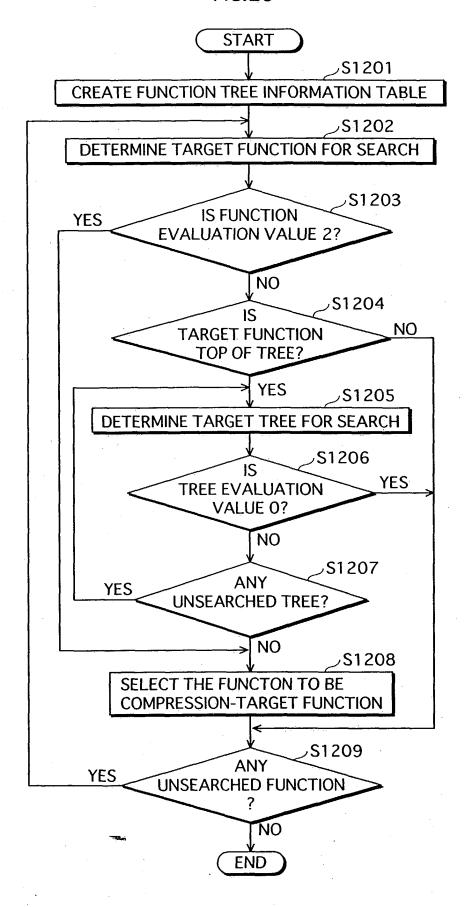


FIG.23



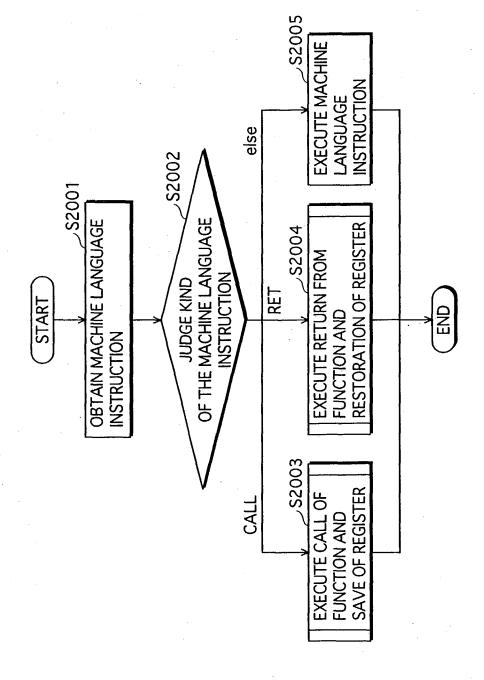


FIG.24

